Containers & Workflow Management Systems

Reproducible computational pipelines with Docker & Nextflow

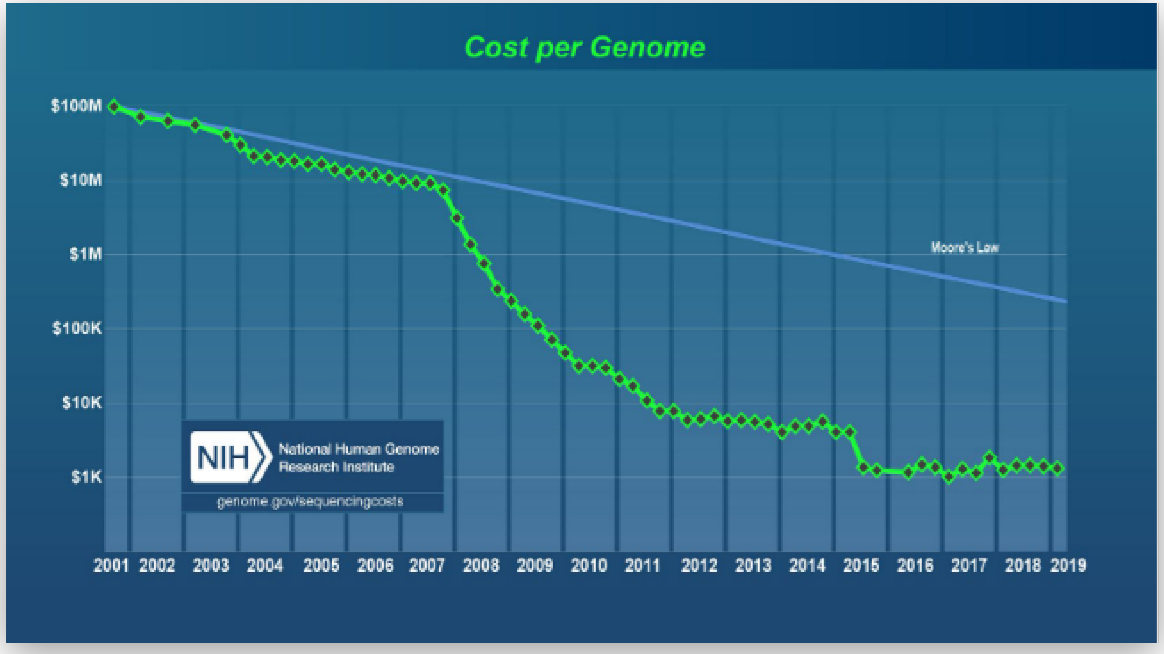
Martin Hölzer, 2022

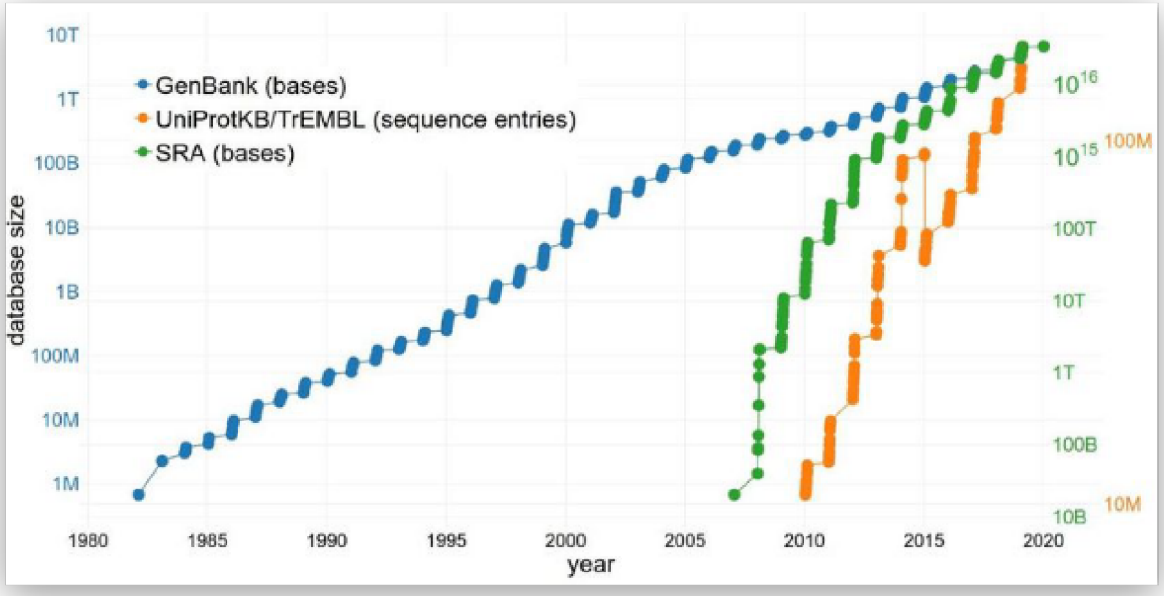
Disclaimer

- limited time: so just some basics/ general ideas
- there are two popular Workflow Management Systems (WMS) in bioinformatics `Snakemake` & `Nextflow`
- they often work w/ containers (prominent: `Docker`)
- easy to start
- but difficult to develop robust workflows
- but once you have them, easy to install, maintain, and use ✊

Two major challenges in computational biology

1. Amount of data





Two major challenges in computational biology

1. Amount of data
2. Reproducibility

Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome

Daniel Garijo[1], Sarah Kinnings[2], Li Xie[3], Lei Xie[4], Yinliang Zhang[5], Philip E. Bourne[3]*, Yolanda Gil[6]*

We estimated the overall time to reproduce the method as 280 hours for a novice with minimal expertise in bioinformatics. The effort included analyzing the paper and the original author's web site and additional materials (data, scripts, configuration files) to understand the details of the method, locating and preparing the codes, finding appropriate parameter settings, implementing the workflows, asking questions to the authors when necessary, and validating the workflows. It should be noted that the authors of the original experiment were available to answer questions (notably Kinnings, the first author). These questions were related to missing configuration parameters, documentation for the proper invocation of the tools, and validation of the outcome of the intermediate steps. Table 1 estimates the time required to reproduce the method and is broken down by major tasks according to our records.

COMMENT | 01 March 2021

Want to track pandemic variants faster? Fix the bioinformatics bottleneck

Tools, rules and incentives are buckling under the flood of coronavirus genome sequences – to help control the pandemic, researchers need new approaches.

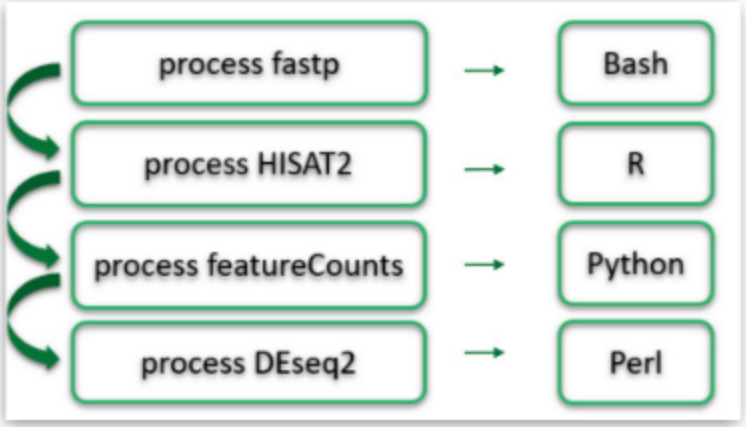
Emma B. Hodcroft ✉, Nicola De Maio, Rob Lanfear, Duncan R. MacCannell, Bui Quang Minh, Heiko A. Schmidt, Alexandros Stamatakis, Nick Goldman ✉ & Christophe Dessimoz ✉

https://www.nature.com/articles/d41586-021-00525-x

What's wrong with computational workflows?

Complexity

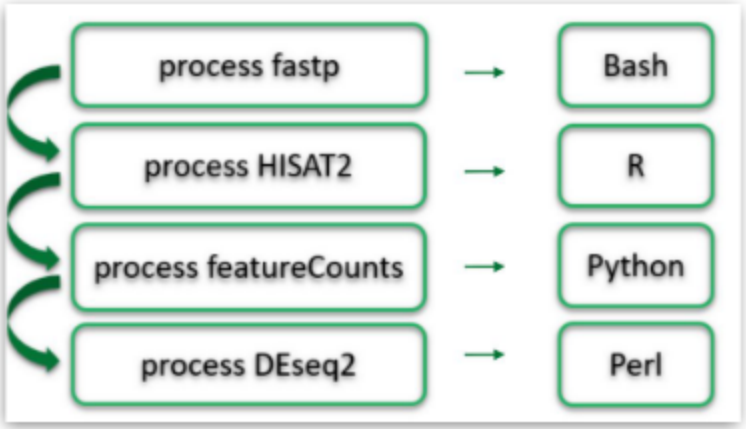
- dozens of dependencies
  - binary tools, compilers, libraries, system tools, ...
- experimental nature of academic software tends to be difficult to install, configure, and deploy
- heterogeneous execution platforms and system architectures



What's wrong with computational workflows?

Complexity

- dozens of dependencies
  - binary tools, compilers, libraries, system tools, ...
- experimental nature of academic software tends to be difficult to install, configure, and deploy
- heterogeneous execution platforms and system architectures



Containers are a game changer

> "A container is a **standard unit of software that packages up code and all its dependencies** so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that **includes everything needed to run an application**: code, runtime, system tools, system libraries and settings."

- transparent build process ('recipe')
- easy to build, share, publish, deploy
- fast instantiation time (~1 sec)
- almost native performance

Containers are a game changer

Example

```
docker run --rm -it -v $PWD:/$PWD -w $PWD staphb/minimap2 minimap2 -h
```

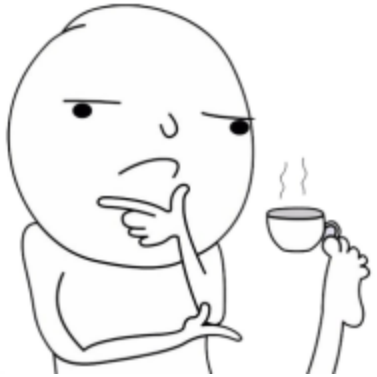
```
(minimap2) ➜  ~ docker run --rm -it -v $PWD:/$PWD -w $PWD staphb/minimap2 minimap2 -h
Unable to find image 'staphb/minimap2:latest' locally
latest: Pulling from staphb/minimap2
7b1a6ab2e44d: Pull complete
34fb90dc1cb1: Pull complete
ab6cbc436bb0: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:c8a3066c506734793953b5d41fd1c69ccfe755ed93039908a40e3f84caacc7e9
Status: Downloaded newer image for staphb/minimap2:latest
Usage: minimap2 [options] <target.fa>|<target.idx> [query.fa] [...]
Options:
```

Containers are a game changer

Example

```
docker run --rm -it -v $PWD:/$PWD -w $PWD staphb/minimap2 minimap2 -h
```

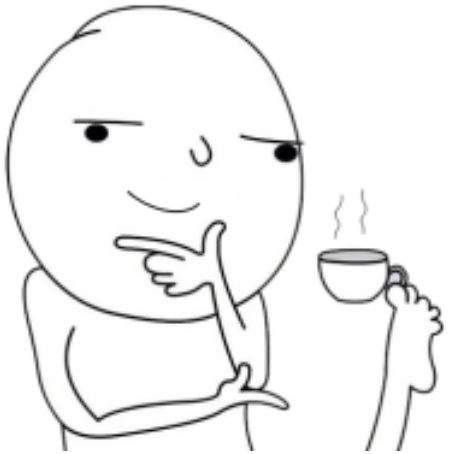
Why not just using Conda?



```
hoelzerm@sebio04:~/git/clean$ conda create -n test -c bioconda spades
Collecting package metadata: done
Solving environment: \
```

Containers are a game changer

- We can install Conda environments into Docker container!
- Allows to easily auto-build containers
  - https://hub.docker.com/
  - https://biocontainers.pro/



Small container example: via `minimap2`

A versatile pairwise aligner for genomic and spliced nucleotide sequences.

JOURNAL ARTICLE

Minimap2: pairwise alignment for nucleotide sequences FREE

Heng Li

Bioinformatics, Volume 34, Issue 18, 15 September 2018, Pages 3094–3100,
https://doi.org/10.1093/bioinformatics/bty191
Published: 10 May 2018 **Article history** ▾

README.md

Download 69k | BioConda install 396k | pypi v2.24 | CI passing

Getting Started

```
git clone https://github.com/lh3/minimap2
cd minimap2 && make
# long sequences against a reference genome
./minimap2 -a test/MT-human.fa test/MT-orang.fa > test.sam
```

Small container example: via `minimap2`

```
# base image
FROM continuumio/miniconda3
#FROM ubuntu:xenial

# install basic libraries and tools
RUN apt update && apt install -y procps wget gzip && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*

# configure conda channels
RUN conda config --add channels conda-forge && \
    conda config --add channels bioconda && \
    conda config --add channels default

# regular conda stuff (w/ fixed tool version for minimap2)
RUN conda install -y minimap2=2.24
RUN conda clean -a
```

Small container example: via `minimap2`

- store the content (the 'recipe') in a file called `Dockerfile` in some folder
- then run:

```
docker build -t mhoelzer/minimap2:2.24 .
```

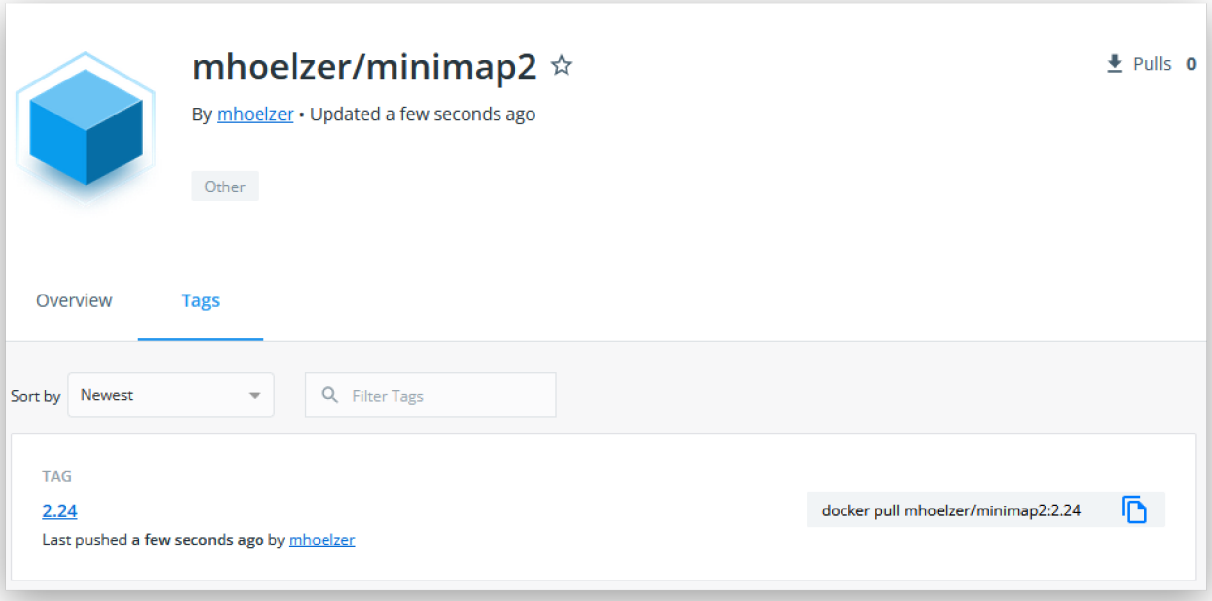
```
(base) ➜  minimap2-docker-testbuild docker build -t mhoelzer/minimap2:2.24 .
Sending build context to Docker daemon  2.048kB
Step 1/5 : FROM continuumio/miniconda3
 ---> ce7d119281a1
Step 2/5 : RUN apt update && apt install -y procps wget gzip &&     apt-get clean &&     rm -rf /var/lib
/apt/lists/* /tmp/* /var/tmp/*
 ---> Using cache
 ---> 21870ffc4bcf
Step 3/5 : RUN conda config --add channels conda-forge &&     conda config --add channels bioconda &&
  conda config --add channels default
 ---> Using cache
 ---> c8b5fdb47a09
Step 4/5 : RUN conda install -y minimap2=2.24
 ---> Using cache
 ---> be15fd29d688
Step 5/5 : RUN conda clean -a
 ---> Using cache
 ---> f725abc59d1d
Successfully built f725abc59d1d
Successfully tagged mhoelzer/minimap2:2.24
```

Small container example: via `minimap2`

- use `docker images` to show images available on your system

```
(base) ➜  minimap2-docker-testbuild docker images | grep mhoelzer/minimap2
mhoelzer/minimap2                                 2.24                f725abc59d1d   About a minute ago   632MB
```

- use `docker push mhoelzer/minimap2:2.24` to push an imagine to a repository



Small container example: via `minimap2`

- to execute a Docker container we have different options
- **in general**: the container we build, push, and pull is a Docker **image**
- we can deploy one or multiple **container**(s) from an **image**

```
# get the image
docker pull mhoelzer/minimap2:2.24

# run a container deployed from the image and start an interactive session
# when the interactive session is stopped, delete the container (keep clean)
docker run --rm -it mhoelzer/minimap2:2.24 /bin/bash
(base) root@had8932h0r82f3j0f2:/ minimap2 --version

# execute a command directly from a deployed container
docker run --rm mhoelzer/minimap2:2.24 minimap2 --version
```

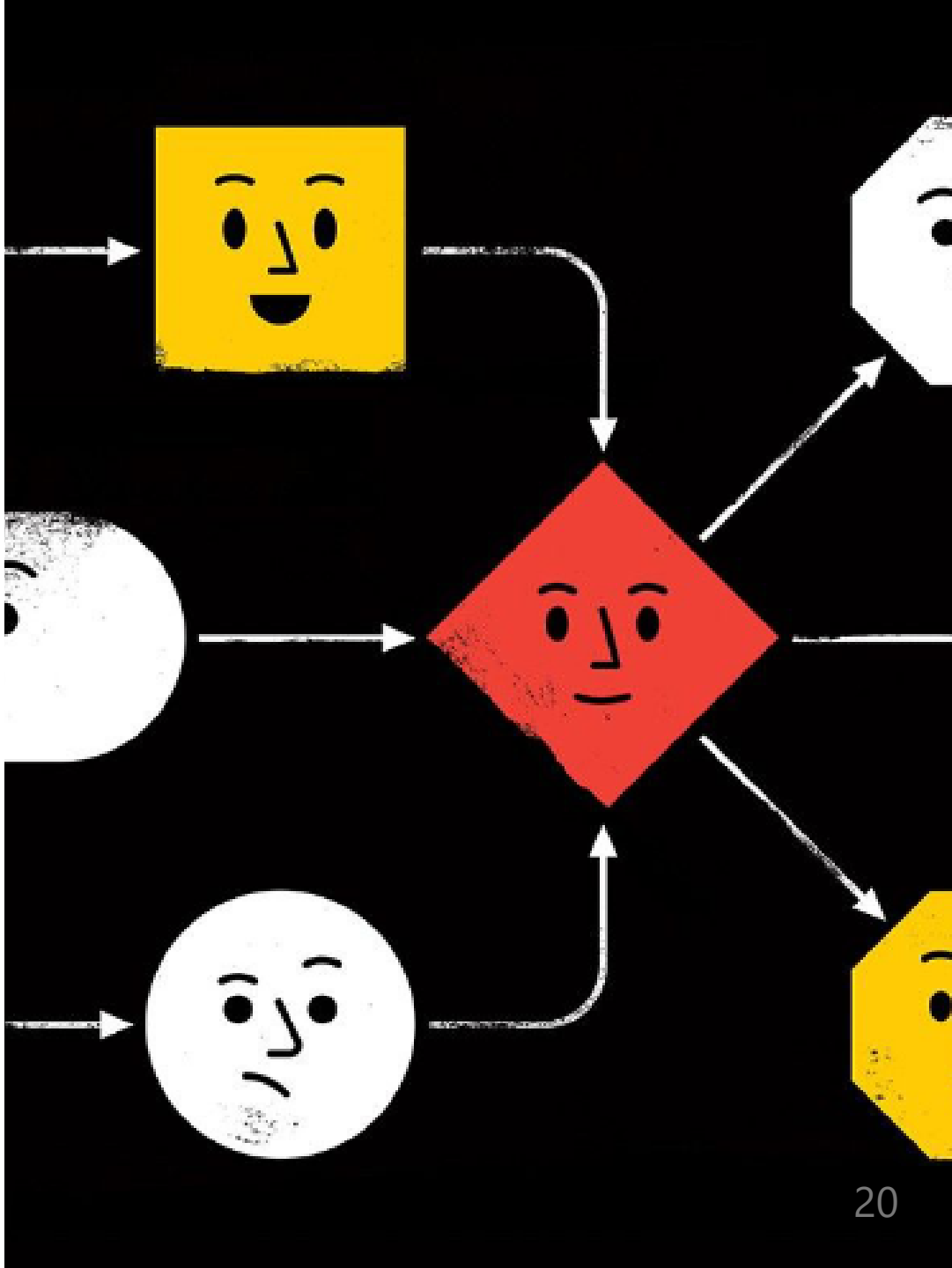
Questions?

Workflow management systems

Workflow: a sequence of operations to complete a process.
WMS: a software that provides an infrastructure to setup, execute, and monitor (scientific) workflows
Bioinformatics WMS: a specialized form of WMS designed to compose and execute a series of computationl steps related to bioinformatics.

WMS

- wrap around command line tools (e.g. encapsulated in Containers) to help with:
  - multi-step workflows
  - software dependencies
  - HPC/ cloud computing
    - scalability, portability
  - save time
    - modularity, maintainable
  - keeping track of what you've done
    - **reproducibility**



WMS

- **Snakemake**
- Nextflow
- Cromwell
- Common Workflow Language

nextflow

COMMON
WORKFLOW
LANGUAGE

Snakemake

- Based on GNU make
- Scripting language is an extension of `Python` — can seamlessly combine `Python` code and workflow commands
- Processes programmed as **rules** with predefined input and output files
- Closely Integrated with `Conda`

```
# install with conda
conda install -n base -c conda-forge mamba
conda activate base
mamba create -c conda-forge -c bioconda -n snakemake snakemake
# install with pip
pip install git+https://github.com/snakemake/snakemake
```

Thank to Maria Trofimova for providing the Snakemake slides!

Snakemake

Example Snakefile and rules

```
rule bwa_map:
    input:
        "data/genome.fa",
        "data/samples/{sample}.fastq"
    output:
        "mapped_reads/{sample}.bam"
    shell:
        "bwa mem {input} | samtools view -Sb - > {output}"
```

see also https://snakemake.readthedocs.io/en/stable/tutorial/short.html

Köster, Johannes, and Sven Rahmann. "Snakemake — a scalable bioinformatics workflow engine." Bioinformatics 28.19 (2012): 2520-2522.

Snakemake

Example: Snakefile and target

- output can be defined via target file in command:

```
snakemake -np mapped_reads/A.bam
```

- or with "all" rule inside Snakefile:

```
rule all:
    input:
        "plots/quals.svg"
```

Snakemake

Wrappers

- allow to use popular tools w/o specifying full tool commands

```
rule samtools_sort:
    input:
        "mapped/{sample}.bam"
    output:
        "mapped/{sample}.sorted.bam"
    params:
        "-m 4G"
    threads: 8
    wrapper:
        "0.2.0/bio/samtools/sort"
```

Snakemake (and Conda actually)

Track dependencies and their versions with environment files.

- Channels: tool storage locations
- Tool versions
- Install from GitHub

```yaml
name: GInPipe
channels:
  - r
  - agbiome
  - defaults
  - conda-forge
  - bioconda
  - anaconda
dependencies:
  - bbmap
  - pip
  - seqkit
  - samtools
  - numpy==1.20.0
  - pysam
  - biopython
  - pandas
  - scipy
  - minimap2
  - pyvcf
  - pip:
    - git+https://github.com/KleistLab/ginpipepy
```

Snakemake

Command Line Interface (CLI)

```
snakemake \
    # path to Snakefile
    --snakefile <snakefilename> \
    # Path to config file
    --configfile <configname> \
    # Number of CPUs to use
    --cores \
    # Working directory
    -d <dirname>
    # Print a graph of executed rules in the workflow
    --dag
    ...
```

WMS

- Snakemake
- **Nextflow**
- Cromwell
- Common Workflow Language

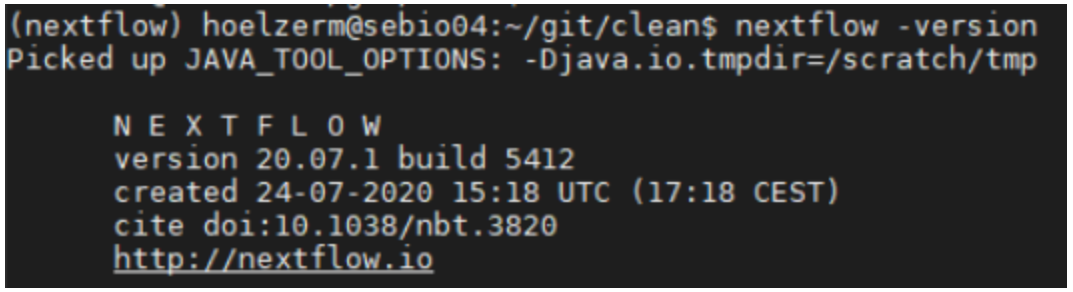
nextflow

COMMON
WORKFLOW
LANGUAGE

Nextflow

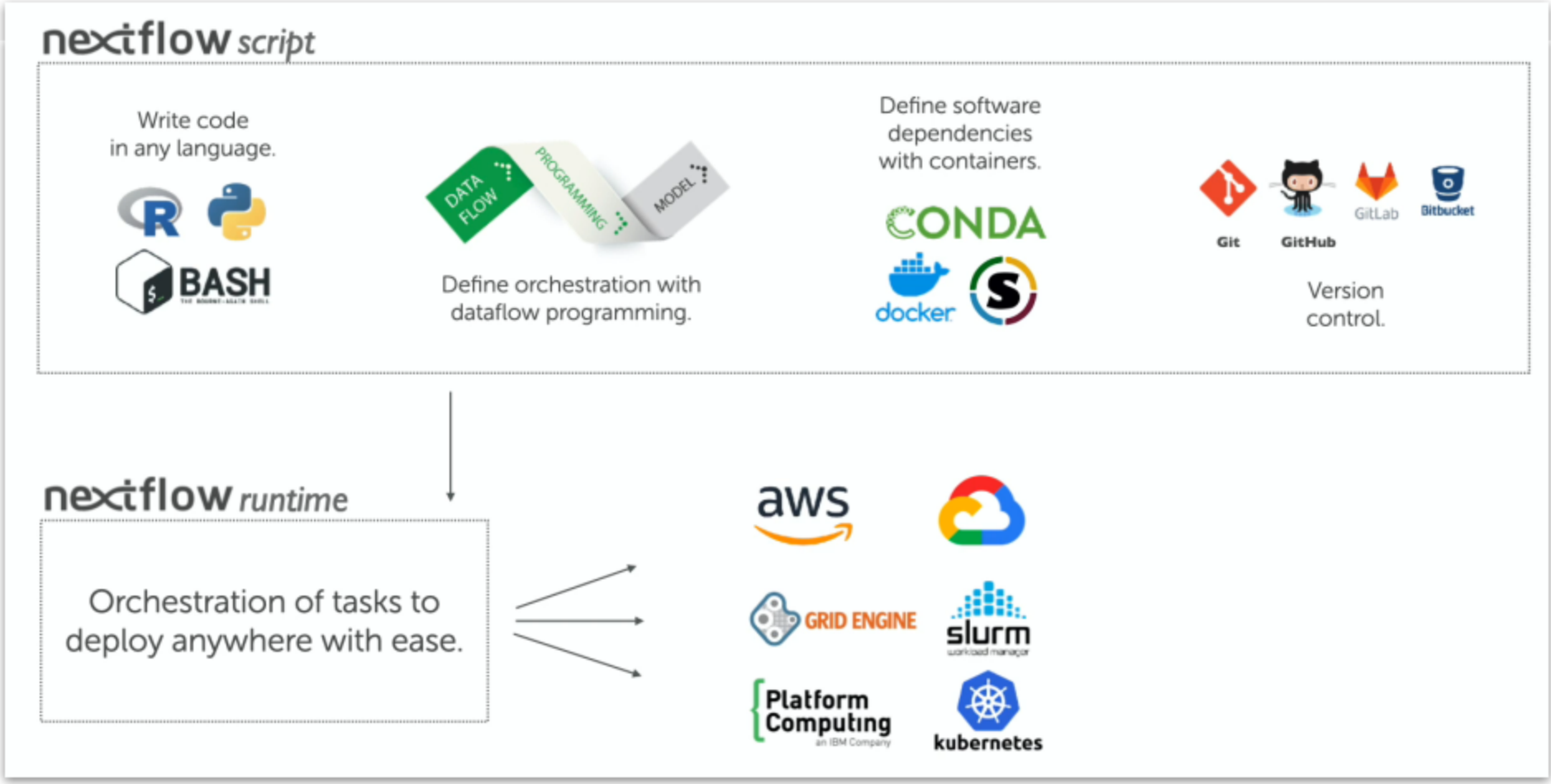
- a framework to run the same pipeline across different platforms
- high level parallelization
- dataflow channel handling (that's the difficult part)
- based on Groovy (which is based on Java), see also https://nextflow.io/

```
# simple install (needs Java!)
curl -s https://get.nextflow.io | bash
# or via conda
conda create -n nextflow -c bioconda nextflow
```



Di Tommaso, Paolo, *et al.* "Nextflow enables reproducible computational workflows." Nature biotechnology 35.4 (2017): 316-319.

Nextflow


nextflow script
Write code
in any language.
R
BASH
DATA FLOW
PROGRAMMING
MODEL
Define orchestration with
dataflow programming.
Define software
dependencies
with containers.
CONDA
docker
Git
GitHub
GitLab
Bitbucket
Version
control.
nextflow runtime
Orchestration of tasks to
deploy anywhere with ease.
aws
GRID ENGINE
slurm
Platform
Computing
kubernetes

Nextflow

DSL2

A major revision of the Nextflow DSL

- Pipeline modularisation
- Component reuse
- Fluent definition of recurrent implementation patterns

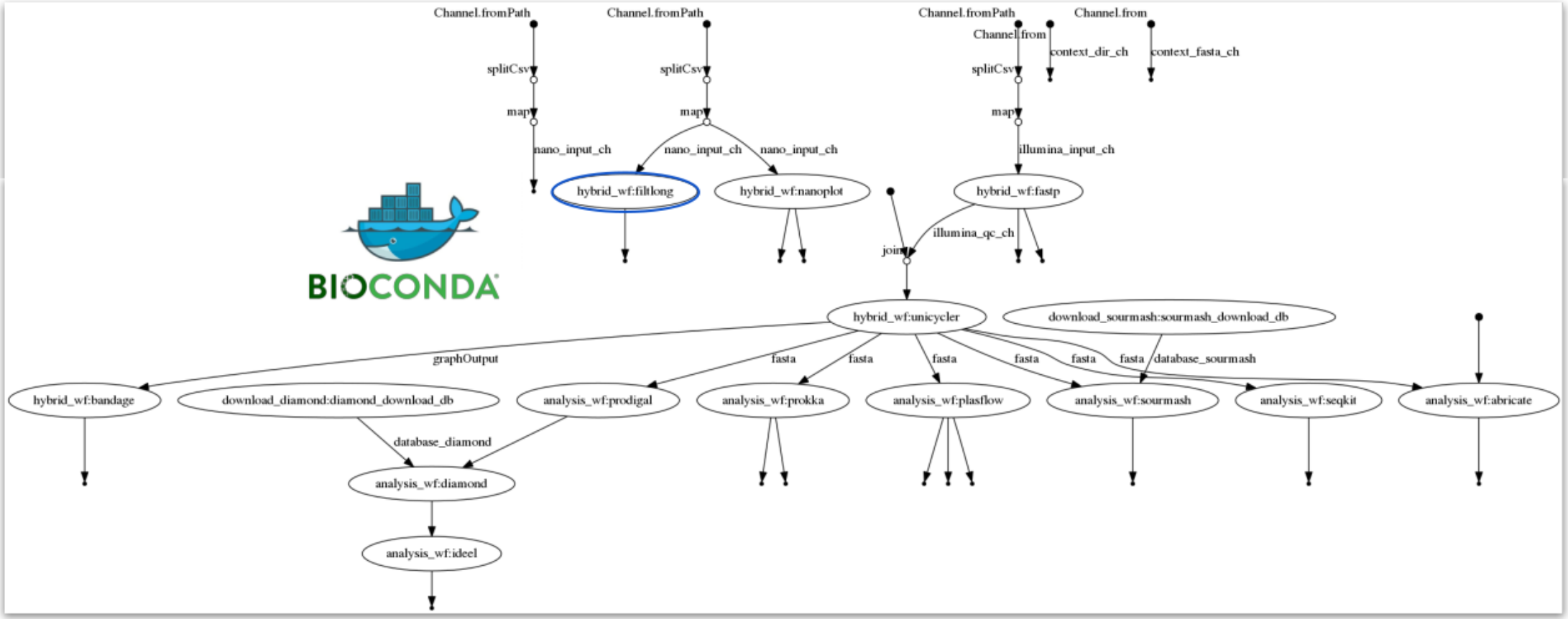
https://bcc2020.sched.com/event/coM5/evolution-of-the-nextflow-workflow-management-system

```
#!/usr/bin/env nextflow
nextflow.enable.dsl=2

if (params.reference) { input_reference_fasta = Channel.fromPath(params.reference) }
if (params.query) { input_query_fasta = Channel.fromPath(params.query) }

process ALIGN {
    container 'mhoelzer/minimap2:2.24'
    publishDir "results", mode: 'copy', pattern: "*.sam"

    input:
    tuple path(query), path(reference)

    output:
    path("${query.simpleName}.sam")

    script:
    """
    minimap2 -ax asm5 ${reference} ${query} > ${query.simpleName}.sam
    """
}

workflow {
    ALIGN(input_query_fasta.combine(input_reference_fasta))
}
```

Full code example and hands-on see https://github.com/hoelzer/nf-minimap2

Nextflow DAG

```
nextflow run main.nf -with-dag dag.png
```



Nextflow workflow report

`[maniac_albattani]` *(resumed run)*

Workflow execution completed successfully!

Run times

26-Aug-2020 09:28:16 - 26-Aug-2020 10:37:38 (duration: **1h 9m 23s**)

68 succeeded | 136 cached

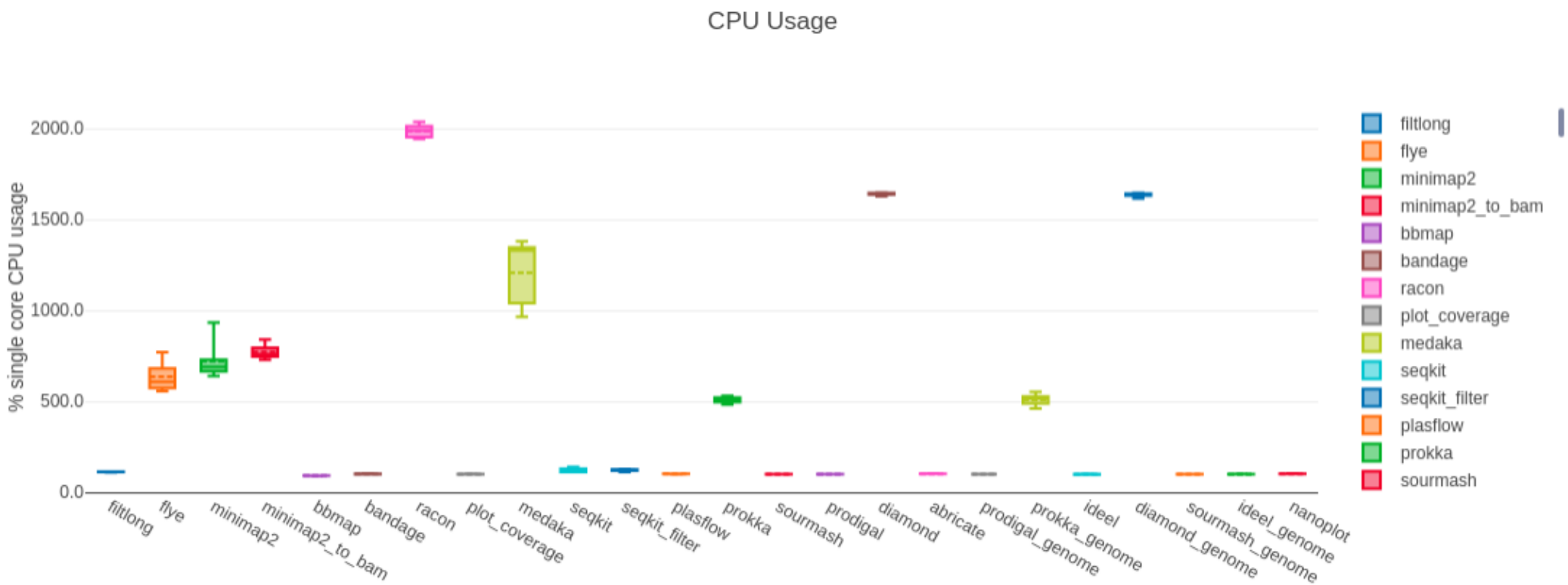
Nextflow command

```
nextflow run main.nf --nano '/home/martin/Downloads/2020-3/*/*.fastq.gz' -profile gcloudMartinPrivate --model r941_min_high_g303 -resume
```

CPU-Hours	177.6 (72.1% cached)
Launch directory	/home/martin/git/wf_reconstruct-strains_prokaryotic
Work directory	/tmp/nextflow-prokaryotic-martin
Project directory	/home/martin/git/wf_reconstruct-strains_prokaryotic
Script name	main.nf
Script ID	efc1ee66ff2007abf90006f13851e56a
Workflow session	dc5f7ef8-ac1e-4077-a89e-a4d3290b7f30
Workflow profile	gcloudMartinPrivate
Nextflow version	version 20.07.1, build 5412 (24-07-2020 15:18 UTC)

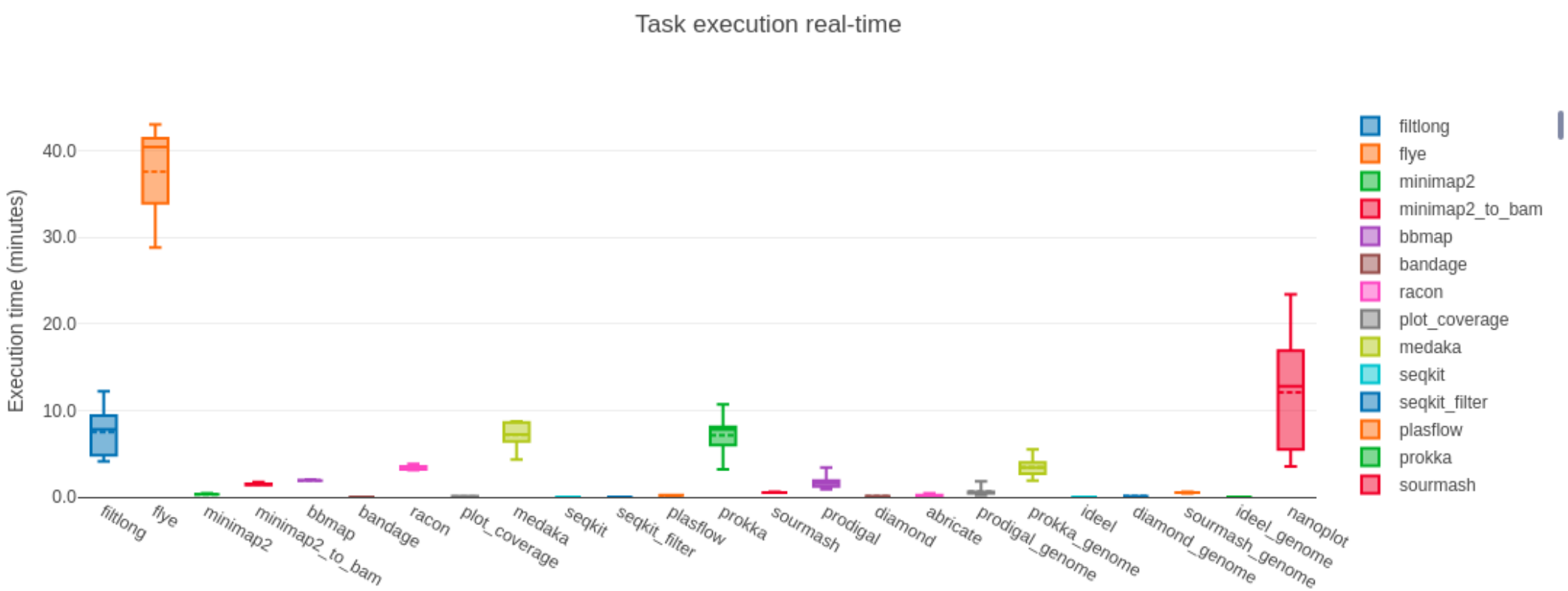
CPU

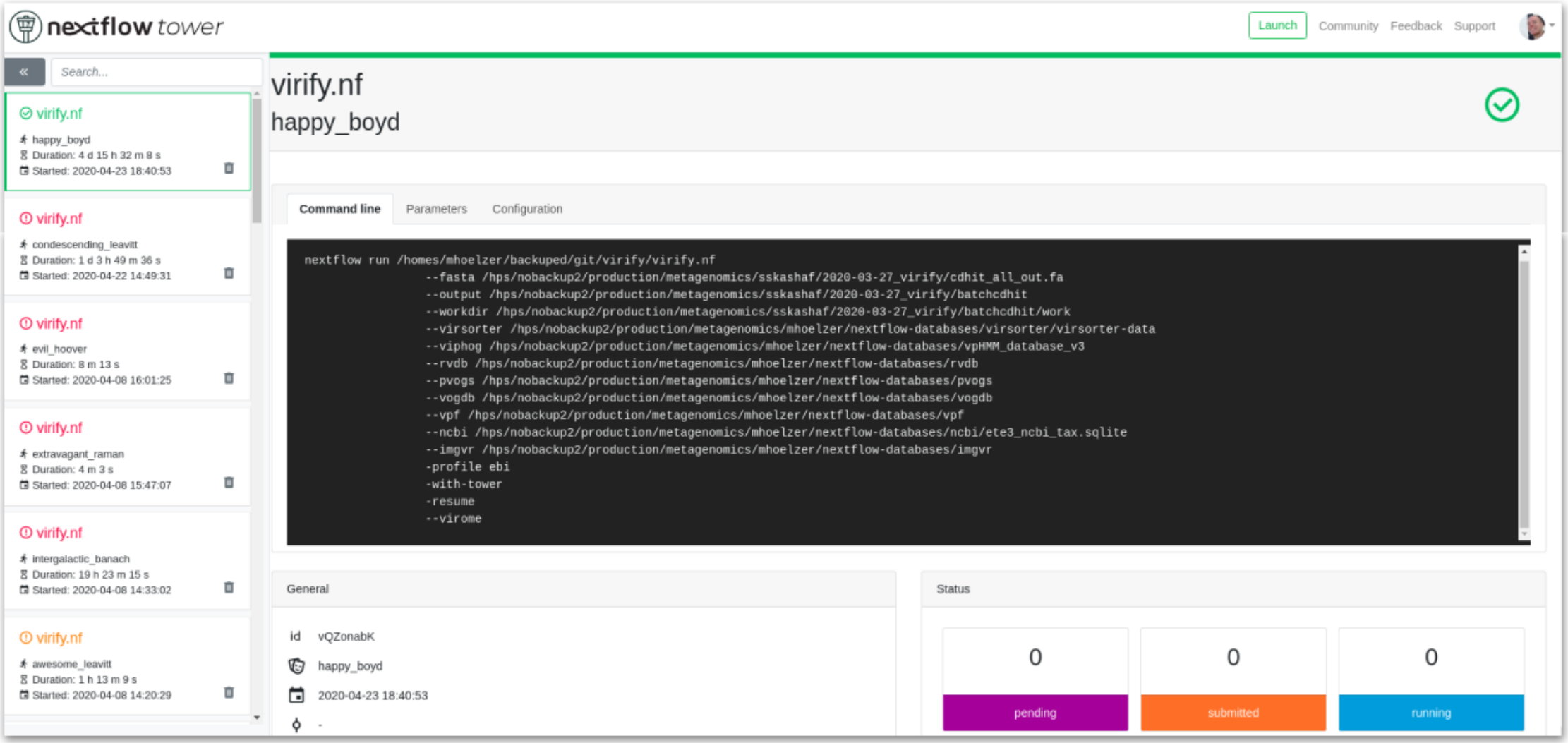
Raw Usage | % Allocated

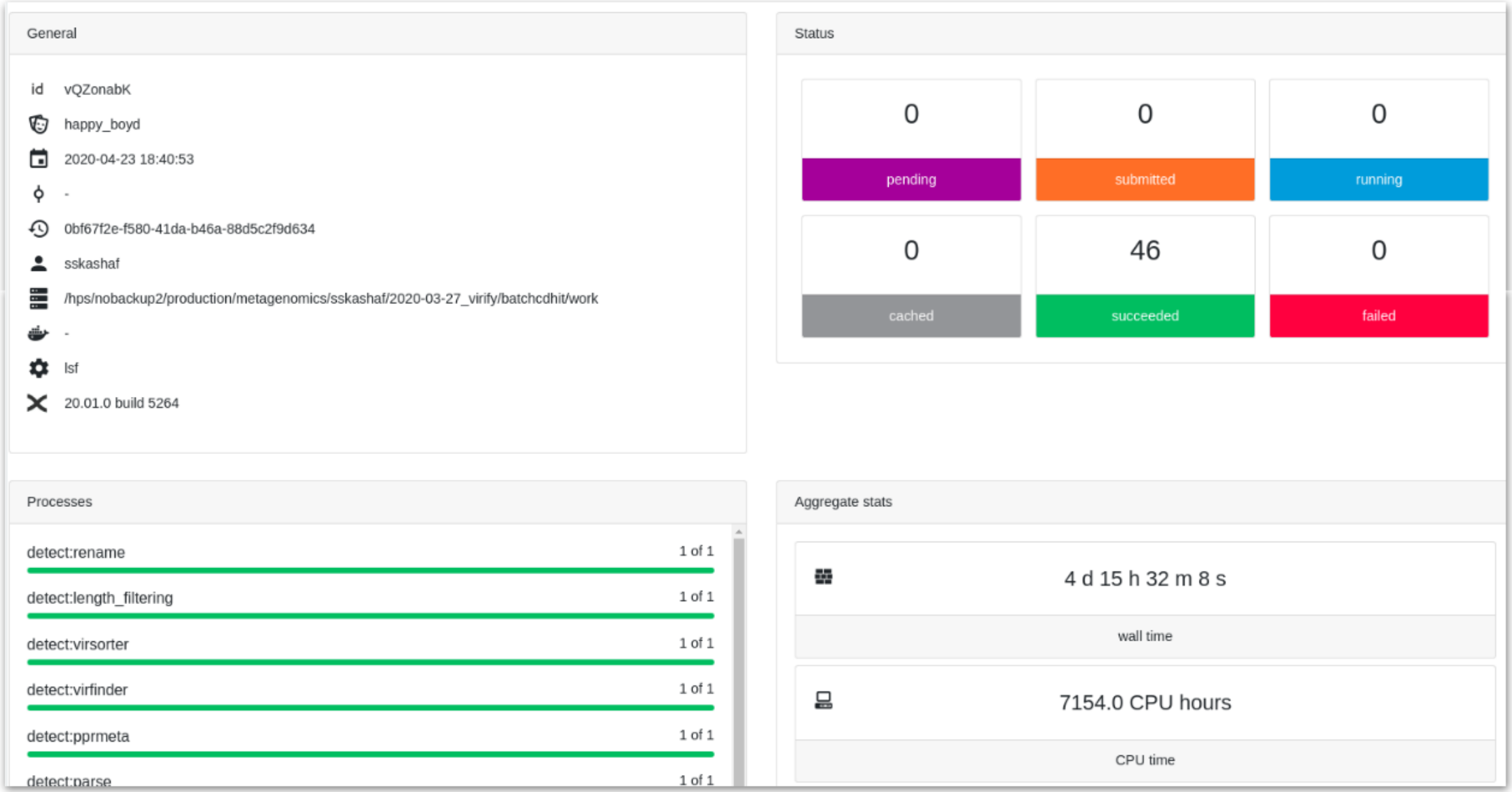


Job Duration

Raw Usage | % Allocated




nextflow tower
Launch
Community
Feedback
Support
Search...
virify.nf
happy_boyd
Duration: 4 d 15 h 32 m 8 s
Started: 2020-04-23 18:40:53
virify.nf
condescending_leavitt
Duration: 1 d 3 h 49 m 36 s
Started: 2020-04-22 14:49:31
virify.nf
evil_hoover
Duration: 8 m 13 s
Started: 2020-04-08 16:01:25
virify.nf
extravagant_raman
Duration: 4 m 3 s
Started: 2020-04-08 15:47:07
virify.nf
intergalactic_banach
Duration: 19 h 23 m 15 s
Started: 2020-04-08 14:33:02
virify.nf
awesome_leavitt
Duration: 1 h 13 m 9 s
Started: 2020-04-08 14:20:29
virify.nf
happy_boyd
Command line
Parameters
Configuration
nextflow run /homes/mhoelzer/backuped/git/virify/virify.nf
--fasta /hps/nobackup2/production/metagenomics/sskashaf/2020-03-27_virify/cdhit_all_out.fa
--output /hps/nobackup2/production/metagenomics/sskashaf/2020-03-27_virify/batchcdhit
--workdir /hps/nobackup2/production/metagenomics/sskashaf/2020-03-27_virify/batchcdhit/work
--virsorter /hps/nobackup2/production/metagenomics/mhoelzer/nextflow-databases/virsorter/virsorter-data
--viphog /hps/nobackup2/production/metagenomics/mhoelzer/nextflow-databases/vpHMM_database_v3
--rvdb /hps/nobackup2/production/metagenomics/mhoelzer/nextflow-databases/rvdb
--pvogs /hps/nobackup2/production/metagenomics/mhoelzer/nextflow-databases/pvogs
--vogdb /hps/nobackup2/production/metagenomics/mhoelzer/nextflow-databases/vogdb
--vpf /hps/nobackup2/production/metagenomics/mhoelzer/nextflow-databases/vpf
--ncbi /hps/nobackup2/production/metagenomics/mhoelzer/nextflow-databases/ncbi/ete3_ncbi_tax.sqlite
--imgvr /hps/nobackup2/production/metagenomics/mhoelzer/nextflow-databases/imgvr
-profile ebi
-with-tower
-resume
--virome
General
id vQZonabK
happy_boyd
2020-04-23 18:40:53
-
Status
0
pending
0
submitted
0
running


General
id vQZonabK
happy_boyd
2020-04-23 18:40:53
-
0bf67f2e-f580-41da-b46a-88d5c2f9d634
sskashaf
/hps/nobackup2/production/metagenomics/sskashaf/2020-03-27_virify/batchcdhit/work
-
lsf
20.01.0 build 5264
Status
0
pending
0
submitted
0
running
0
cached
46
succeeded
0
failed
Processes
detect:rename 1 of 1
detect:length_filtering 1 of 1
detect:virsorter 1 of 1
detect:virfinder 1 of 1
detect:pprmeta 1 of 1
detect:parse 1 of 1
Aggregate stats
4 d 15 h 32 m 8 s
wall time
7154.0 CPU hours
CPU time

task_id ▲	process	tag	hash	status	exit	container	native_id	submit
⊕ 1	detect:rename		a1/647552	SUCCEEDED	0	/hps/nobackup2/singularity/sskashaf/mhoelzer-python3_virify-0.1.img	4393336	2020-04-23 18:41:24
⊕ 4	detect:virsorter		3b/c55382	SUCCEEDED	0	/hps/nobackup2/singularity/sskashaf/quay.io-biocontainers-virsorter-1.0.6--pl526h516909a_1.img	4393369	2020-04-23 18:42:27
⊕ 6	detect:parse		3a/8454e0	RUNNING	-	/hps/nobackup2/singularity/sskashaf/mhoelzer-python3_virify-0.1.img	5088530	2020-04-24 05:46:21
⊕ 7	detect:restore		06/993dfd	SUCCEEDED	0	/hps/nobackup2/singularity/sskashaf/mhoelzer-python3_virify-0.1.img	5088912	2020-04-24 05:46:46
⊕ 8	detect:restore		9f/77697e	RUNNING	-	/hps/nobackup2/singularity/sskashaf/mhoelzer-python3_virify-0.1.img	5088899	2020-04-24 05:46:46
⊕ 9	detect:restore		f3/a246d0	SUCCEEDED	0	/hps/nobackup2/singularity/sskashaf/mhoelzer-python3_virify-0.1.img	5088927	2020-04-24 05:46:47
⊕ 10	annotate:prodigal		13/98667d	SUCCEEDED	0	/hps/nobackup2/singularity/sskashaf/mhoelzer-prodigal_viral-0.1.img	5089046	2020-04-24 05:46:56
⊕ 11	annotate:prodigal		d3/dadce6	PENDING	-	/hps/nobackup2/singularity/sskashaf/mhoelzer-prodigal_viral-0.1.img		-
⊕ 12	annotate:prodigal		cc/4f2af5	SUBMITTED	-	/hps/nobackup2/singularity/sskashaf/mhoelzer-prodigal_viral-0.1.img	5089077	2020-04-24 05:46:57
⊕ 13	annotate:hmmscan_viphogs		73/e42047	SUCCEEDED	0	/hps/nobackup2/singularity/sskashaf/mhoelzer-hmmscan-0.1.img	5089590	2020-04-24 05:47:31
⊕ 15	annotate:ratio_evalue		46/6d861c	RUNNING	-	/hps/nobackup2/singularity/sskashaf/mhoelzer-python3_virify-0.1.img	5106758	2020-04-24 06:09:26
⊕ 16	annotate:annotation		83/3854cc	SUCCEEDED	0	/hps/nobackup2/singularity/sskashaf/mhoelzer-annotation_viral_contigs-0.1.img	5107201	2020-04-24 06:09:41
⊕ 17	annotate:plot_contig_map		a4/f317bc	SUCCEEDED	0	/hps/nobackup2/singularity/sskashaf/mhoelzer-mapping_viral_predictions-0.2.img	5107451	2020-04-24 06:09:53
⊕ 18	annotate:assign		f4/e61652	SUCCEEDED	0	/hps/nobackup2/singularity/sskashaf/mhoelzer-assign_taxonomy-0.1.img	5107506	2020-04-24 06:09:55
⊕ 19	annotate:hmmscan_viphogs		8d/586b82	SUCCEEDED	0	/hps/nobackup2/singularity/sskashaf/mhoelzer-hmmscan-0.1.img	5109349	2020-04-24 06:11:06

Questions?

Nextflow: example code and try yourself

See: https://github.com/hoelzer/nf-minimap2

Further reading & helpful resources

Container

- https://www.happykhan.com/posts/dark-secret-about-containers
- https://github.com/sib-swiss/containers-introduction-training

Snakemake

- Snakemake documentation: https://snakemake.readthedocs.io/en/stable
- Snakemake on SLURM: https://bihealth.github.io/bih-cluster/slurm/snakemake/

Nextflow

- The story of Nextflow: Building a modern pipeline orchestrator: https://elifesciences.org/labs/d193babe/the-story-of-nextflow-building-a-modern-pipeline-orchestrator
- Extensive training material (likely DSL1, but can be adapted to DSL2): https://training.seqera.io
- Some latest training material: https://seqera.io/nextflow/learn
- EBI NF course material: https://www.ebi.ac.uk/training/online/courses/nextflow